NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

TESTING THE FORENSIC INTERESTINGNESS OF IMAGE FILES BASED ON SIZE AND TYPE

by

Raymond M. Goldberg

September 2017

Thesis Advisor: Neil Rowe
Second Reader: George Dinolt

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2017	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE TESTING THE FORENSIC INTERESTINGNESS OF IMAGE FILES BASED ON SIZE AND TYPE		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Raymond M. Goldberg			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

In this thesis, we investigate the relationship between the size and type of a file and its forensic usefulness. We investigate GIF, MP3, MP4, PNG, and JPEG files found in a large collection called the Real Drive Corpus, and the files' classification as software-based, entertainment-based, or personal. Results of these experiments were compared to prior work to find interesting files. Results show that the previous experiments were effective at marking interesting files as interesting, but there were still a lot of uninteresting files that were marked as interesting. Also, the results do not show a correlation between the interestingness of a file, its type, and its size.

14. SUBJECT TERMS Real Drive Corpus, scanning, white listing, known files database			**15. NUMBER OF PAGES** 45
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

TESTING THE FORENSIC INTERESTINGNESS OF IMAGE FILES BASED ON SIZE AND TYPE

Raymond M. Goldberg
Second Lieutenant, United States Army
B.S., Florida Southern College, 2016

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN CYBER SYSTEMS AND OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
September 2017

Approved by: Dr. Neil Rowe
Thesis Advisor

Dr. George Dinolt
Second Reader

Dr. Dan Boger
Chair, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In this thesis, we investigate the relationship between the size and type of a file and its forensic usefulness. We investigate GIF, MP3, MP4, PNG, and JPEG files found in a large collection called the Real Drive Corpus, and the files' classification as software-based, entertainment-based, or personal. Results of these experiments were compared to prior work to find interesting files. Results show that the previous experiments were effective at marking interesting files as interesting, but there were still a lot of uninteresting files that were marked as interesting. Also, the results do not show a correlation between the interestingness of a file, its type, and its size.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

GIF	Graphics Interchange Format
JPEG	Joint Photographic Experts Group
LSH	Locality Sensitive Hashing
NSRL	National Software Reference Library
PDF	Portable Document Format
PNG	Portable Network Graphic
RDC	Real Drive Corpus

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Thank you to my parents for helping me get through this. Also, thank you to Dr. Rowe and Dr. Dinolt for sticking with me and editing this paper. Finally, to the Rods, thank you for keeping me (relatively) sane.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In today's world, technology is more widespread than ever before. People store their business transactions, electronic communications, and personal pictures on their personal computers or smart phones. While this has made it easier for people to create, share, and store information, it has also created a new challenge for law-enforcement and intelligence communities. Hard drives can hold massive amounts of data, most of which is uninteresting to investigators. For example, the computer's operating system (OS) will be on the hard drive, as well as any other installed software such as Flash Player, Adobe Reader, Google Chrome, etc. An analyst would have to review 100% of the contents of the hard drive manually to find the 1–5% (approximate) of files that actually contain data of interest. As a further example, if the Federal Bureau of Investigation (FBI) conducted a raid on an organized crime group, there is a good chance that the group stored sensitive business information on a computer hard drive. However, the group could have hidden these few interesting files among the many uninteresting software files, thus creating a "finding a needle in a haystack" problem.

First, the term "interesting file" needs to be defined. Neil Rowe defines interesting files as,

> those [files] that are relevant in criminal or intelligence investigations. Usually, this means they are 'probative' about a human subject of investigation, or prove something about them that can be used in an investigation. This includes documents and media created by a subject, and does not include downloads, streams, or software. (N. Rowe, personal communication, September 12, 2017)

For this thesis, interesting files that were found during experimentation were entirely personal, not software-based or entertainment-based. In summary, for a file to be considered interesting, they had to be directly attributable to the user or their friends/family, or they had to be evidence of a crime.

Because many hours would be needed to look through the entire drive to find the interesting files described above, researchers have tried to automate this process with software. The goal for developing such software is not to eliminate all of the

uninteresting files so that the analyst only needs to look at the interesting ones, but to eliminate as many of the uninteresting files as possible without eliminating any interesting ones (labeling key evidence as uninteresting would be very bad for a case). While there are many different ways for identifying a file as interesting or not, they are all mostly generic rules for all files. For example, one can compare a hash of a file to a "whitelist" of known benign files (Cawathe, 2012; Rowe, 2015; Ruback, Hoelz, & Ralha, 2012). If it matches, then the file is regarded as uninteresting (Cawathe, 2012, p. 591). For example, Rowe (2015) created a program that tests file metadata against several criteria for uninterestingness and interestingness. Such rules included files that were created within a very short period of time from one another, files that were created during busy weeks, and files in which a large amount of other files in the directory are uninteresting.

While these rules can get rid of a large percentage of uninteresting files, they can also mistakenly get rid of interesting files as well (Rowe, 2015). For example, if an enemy wanted to hide sensitive documents on a computer so that file-scanning would not see them, then they only need to hide the documents inside a regularly used Graphics Interchange Format (GIF) file that is in an uninteresting directory. The bulk of GIFs on a hard drive are used by software for various reasons, usually for their Graphical User Interface (GUI). While this could make it tempting to expel all GIFs as uninteresting, it could also make them tempting vehicles for hiding information in plain sight using stenography for example.

Because of this problem, it is important to test the interestingness of GIF and other image files. Chapter II will delve into the previous ways that other people have tried to help with scanning hard drives, to give context to the new rules tested here. Chapter III will describe the attributes scanned in image files. The remaining chapters will describe the experimental methodology, the results of the experiment, and conclusions that were derived from the results.

II. BACKGROUND INFORMATION

It is important to understand which files on a hard drive are forensically "interesting" and which are "uninteresting." For criminal investigations, analysts mainly look for files that contain evidence for a case. This could include bank statements, photographs, emails, audio files, etc., that show a person committing a crime, or at the very least being connected to people who committed a crime. For intelligence investigations, though, the focus is on finding information that could be used to exploit targets. For instance, intelligence analysts are more interested in finding any files that suggest what a particular target's interests are. These could be a list of websites they go to, where they like to shop, and what do they like to do in their leisure time. Also, it is important to find out what people the targets connect to. This information could be found anywhere, but emails, image files, and videos would be good places to look. Finally, it could be important for an analyst to find travel details. For instance, evidence suggests that if a target went to Turkey for a week, the investigative unit could start looking for any potential links between the target and other persons of interest in that area. For the purposes of this thesis, if a file fits into any of the categories listed above, then it is considered interesting. For these experiments, forensic interestingness was binary, yes or no. A file that was "somewhat interesting," "moderately interesting," or "definitely interesting" to an investigator was considered as "interesting" to err on the side of caution. For this thesis, only files that are directly associated to a person were considered interesting. Such files consisted of: personal pictures, homemade video, and homemade audio. While personal text documents, Excel spreadsheets, or presentation slides would also be interesting to an investigation, these types of files were not investigated in this thesis. Any entertainment-based files of games, stores, hotels, etc. were considered uninteresting. Finally, any software-based files were also considered uninteresting, unless they showed some evidence of a crime (such as downloading illegal software). However, none of the software-based files showed any illegal activity during these experiments. In general, if a file was of a type other than personal (such as entertainment-based or software-based), then there was little chance of it being labeled as interesting.

While there has not been any previous work on detecting the interestingness of image files in particular, there have been many papers written on ways to improve file-scanning overall. Ruback et al. (2012), for example, came up with a way to remove more uninteresting files from the hard drive while simultaneously decreasing the number of files that have to be scanned in the known files database (KFDB) (Ruback et al., 2012). The standard approach to removing uninteresting files from the list that has to be looked at by an investigator is to have the scanning software take the hash of the files and look for it in a KFDB, or a "whitelist" (Chawathe, 2012). If the hash of the file in question matches a hash in the KFDB, then the file is unlikely to contain the unique information an investigator seeks, and the investigator does not need to look at it.

To cover the sheer volume of modern software on hard drives, however, this approach must store large numbers of hashes. Ruback et al. (2012) argues that KFDBs are getting so large that they are difficult to implement efficiently since it takes a long time to search for a particular hash. They also note that not all countries or geographical regions have the same files on their computers. Because of these two issues, Ruback et al. (2012) developed a way to use data mining to construct hash-set subsets for KFDBs depending on the country or area that the hard drive was used in. For example, if a hard drive is brought in for investigation from Iraq, and the owner of the hard drive is known to have been in Iraq all of their life, then the scanning software will only scan the drive for specific hashes that are known to be in Iraqi hard drives. Testing these methods yielded a 30.69% increase in filtering results (accuracy) and a 51.83% decrease in the size of the hash-set database being used (Ruback et al., 2012).

A problem with cryptographic hashing is that one small change in the file changes its hash. If someone knew that an investigative organization was comparing hash values to KFDBs, they could make small changes to many uninteresting files, and the scanning-software would label them as interesting thus increasing the amount of work for the analyst (Chawathe, 2012). Furthermore, trying to match files is a binary (match/ no match) operation. If the scanning software could say whether the mismatch was close, then the human analyst could make a judgment call as to whether or not the file is worth looking at. This "partial matching" problem is addressed in Chawathe (2012) which

proposes piecewise signature matching as well as a technique called Locality Sensitive Hashing (LSH) (Chawathe, 2012). This technique can help analysts sift through near-miss files to label some as uninteresting so the analyst does not have to be bothered with them.

Finding near-miss files was also examined by Kornblum (2006) who made a hashing program called "ssdeep." Ssdeep used a mix of piecewise and traditional hash algorithms to create a Context Triggered Piecewise Hash (CTPH) that grades similarity on a scale of 0–100. Later, Long and Guoyin (2008) modified ssdeep to be more accurate at finding similar files.

White-listing to eliminate uninteresting files does not guarantee that the investigator will not get overwhelmed with data, unfortunately. For example, the total number of files in the Real Drive Corpus (a collection of hard drives legally obtained from around the world and stored at the Naval Postgraduate School) is 262.7 million (Rowe, 2015). Eliminating files with hashes in the National Software Reference Library (NSRL) (a large Known Files Database) can reduce this, but this is still a large number of files. To further decrease the number of files, other file properties besides the hash must be examined.

Rowe (2015) addresses this problem by implementing nine methods to test files for their uninterestingness. If the file was identified by two methods as "uninteresting," then it was marked for elimination. However, if a file was identified as "interesting" by at least one of six specialized methods, then the file was re-designated as "interesting." One method looked for hash values that were found only once in the Corpus, but whose paths has a much more common hash value, giving the impression that the file was trying to disguise itself with a benign file path. Another method looks for common hashes that are found in a hard drive but are found on a file that has a different name than the usual one. For instance, if the common file normalFile.txt has a hash of 1234, but the scanning software finds a file by the name of interestingFile.txt that has the same hash, then that would be a sign of "deliberate renaming" (Rowe, 2015). Method three looks for files that are created at unusual times compared to the rest of the files in their directory. Method four looks for inconsistencies between the file extension (e.g., .txt, .pdf, .BAT) and its

header analysis (“magic number” analysis). Method five looked for files with atypical sizes. An example would be if all files in a drive were supposed to have 64-byte hash values, but a few files have 56-byte hashes. Finally, method six looks for file extensions or directories that are specifically known to be interesting by manual designation. Using these methods alongside hash-checking with NSRL led to a 77.4% decrease in the number of files that had to be checked by analysts (about 59.5 million files were left to be checked), while keeping the percentage of interesting files from being mistakenly marked as uninteresting down to 0.18% (Rowe, 2015).

III. IMAGE FILE FORMATS

When scanning a computer hard drive, many kinds of pictures are found. Digital images are not only created by humans for entertainment purposes, but also for software Graphical User Interfaces (GUIs). Many of these images are stored as Graphics Interchange Formats, or GIFs. This includes most buttons, arrows, and icons a user clicks on while operating Windows 10. GIFs were created by Steve Wilhite in 1987 (William, 2016) as a way to transfer images and store them online for the company he was working for, CompuServe Inc. (Fileformat, n.d.). GIF files use lossless compression with the LZW algorithm. Compression is important for image files because they can be big. Since GIFs use lossless compression, an editor can resave the image as many times as needed without fearing degradation of image quality.

GIFs can store multiple images inside one file, and then show all of those images in sequence as of the 1987 revision (known as GIF87a). However, if you speed up the transition rate enough, then the slide show turns into a movie clip (William, 2016). The 1989 revision (GIF89a) allowed creators to show text on top of the pictures.

A useful classification splits them into software-based GIFs, entertainment-based GIFs, and personal GIFs. Personal GIFs rarely incorporate more than one picture per file. However, nearly all GIFs that are used for entertainment purposes (like with online memes or emojis), and some GIFs associated with software, contain multiple images. Entertainment-based and personal GIFs may have value in an investigation because they provide data about user interests. Also, if the scanning software finds a personal GIF in a directory, there are likely other user-created files in the same directory (such as Word files or Excel spreadsheets). Software-based GIFs can almost always be flagged as uninteresting to save the analyst some time and effort.

To help scanning software discriminate between different kinds of GIFs, the most notable difference was file size: Software-based GIFs were typically much smaller than entertainment-based GIFs. For instance, of the nine GIFs from the Corpus that we were fairly certain were software-based GIFs (see Figure 1), seven were less than 10 kilobytes

(KB), one was 91 KB, and one was 1,449 KB. Nine GIFs from an Internet site called giphy.com that we were certain were entertainment-based GIFs had sizes in KB of 204, 291, 431, 607, 735, 982, 1,456, 4,604, and 4,822.

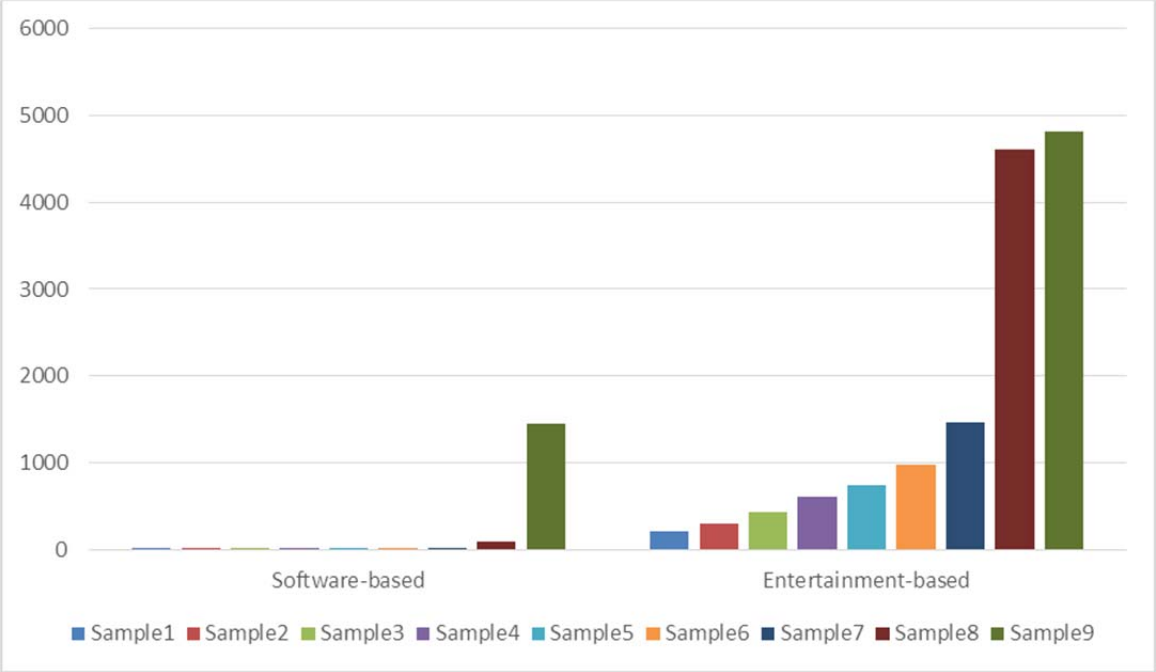

Figure 1. Comparison of Entertainment-Based and Software-Based GIFs

File differentiation based on size could also prove useful for other common image file formats like JPEG and PNG. The Portable Network Graphic (PNG) format was developed to improve color support over the older GIF format, and also to get around the copyright laws that restrict the use of GIF formats (FileInfo, 2017b). PNG files use lossless compression schemes similar to GIF files, but the PNG format can handle color transparency in a way that GIF files cannot. For instance, GIF files can only store colors that do not require any transparency data (such as fading), but PNG files can have an "8-bit transparency channel," (FileInfo, 2017b) that is able to store such data. Also, PNG files cannot be animated like GIF files (FileInfo, 2017b).

A file is designated as a JPEG if it uses the compression format that was developed by the Joint Photographic Experts Group, which is the format's namesake.

JPEGs are Unlike GIFs and PNGs, JPEG files use lossy compression (FileInfo, 2017a), which means some data is lost every time the image needs to be resaved. Finally, the JPEG format is used primarily for storing pictures and graphics for software GUIs (FileInfo, 2017a).

The sizes of 10 PNG files and 10 JPEG files that were randomly picked from the test-batches used for the experiments in this thesis are shown in Figure 2. All of the sample files used in these figures were also regarded as interesting by Rowe's prior tests on the RDC. The JPEG files are usually much larger than the PNG files, of which all but one file was under 1 KB in size.

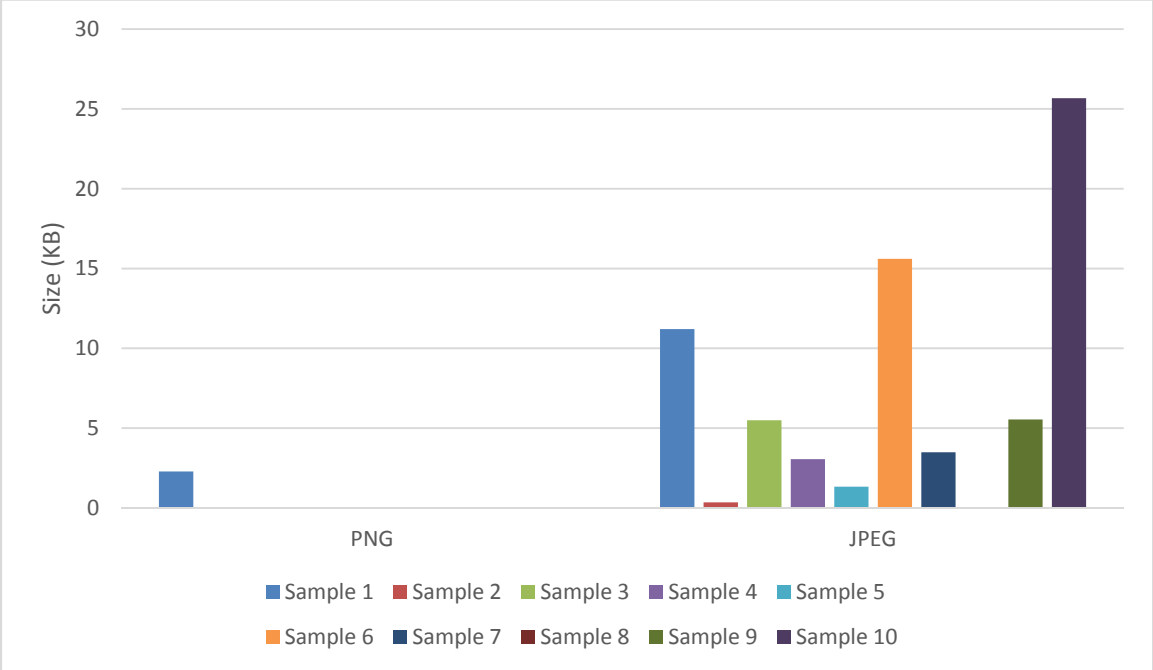

Figure 2. Interesting PNG and JPEG Sample Sizes

In addition to the PNG and JPEG formats, this thesis will also test MP3 and MP4 files. The MP3 format is used for storing audio files, such as music and audiobooks (FileInfo, 2016b), while the MP4 format is used for storing video files (FileInfo, 2016a). Both formats were developed by the Moving Picture Experts Group (FileInfo, 2016a) (FileInfo, 2016b), and they are popular formats for storing their respective data. Because

of their popularity, it was concluded that testing for any connections between file type, size, and interestingness would be worth investigating.

IV. DESCRIPTION OF METHODOLOGY

Our experiments tested the interestingness of a file in relation to its size. Experiments were carried out in three phases: test setup, data transfer, and data testing. The setup phase was responsible for finding metadata for files that were deemed interesting or uninteresting (depending on the sub-experiment) of a specific file type (GIF, JPEG, etc.) from the previous examinations, and then taking a random subset to run the tests on. The transfer phase extracted the actual files referenced in the metadata and copied them to a testing environment. The testing phase manually inspected each file and determined whether the file was truly interesting or uninteresting. Further details of each step in the experiment are given below.

A. TEST SETUP

Setup consisted of accessing the Real Drive Corpus (RDC) and choosing the files for testing. We used the fiwalk14d directory, which had all three files that we would need for the experiment. This directory contained metadata from the state of the corpus in late 2014. While there was also the fiwalk17a directory available for consideration, due to some missing data and files, we decided to go with fiwalk14d.

Once access to the fiwalk14d directory was attained, the next step would be to identify files in the random sample as interesting and uninteresting by Dr. Rowe's previous experiment. For this, we did two database-style joins of the data with lists of hashcodes, one a list of interesting hashcodes and one a list of uninteresting hashcodes.

The next step was to find the particular image files that we wanted to study. To accomplish this, we used the "grep" command to search for the desired file extensions (with or without capitalization) (.gif, for example). The command for finding interesting GIF files was:

```
Grep "\.gif|" join_augmented_hashdata_rdc_final_hahscodes.out
>GREPinterestingGIFs.txt
```

This found many files. To enable manual inspection, we took a random sample by utilizing a python program called randchoose, which was developed by Rowe. Using randchoose reduced the number of data files that had to be transferred from the RDC to my computer. It should be noted that for both the uninteresting and interesting MP4 randchoose files we used a sampling ratio of 0.1; but, for every other file type, we used 0.01 as the ratio because the MP4 files found by grep were a lot less than the other grep files.

For study of GIF files, we chose 100 files that were declared to be interesting from Rowe's previous experiments, and 100 files that were supposed to be uninteresting. For the other file types, we only chose 50 of each because they were not the main file type to be investigated by this thesis. To subsample these files further, we used a Google Chrome plug-in called Randomgen, by VaughnGH.

B. DATA TRANSFER

The Data Transfer phase used the file metadata from the previous phase (like that in Figure 1) to get the actual files from the RDC. Files in disk images in the RDC can be accessed and scanned by a variety of forensic tools. We used open-source software called The Sleuth Kit, which is "a library and collection of command line tools that allow you to investigate disk images" (Carrier, n.d.). It is a tool set that is available for free for people interested in digital forensics. To extract a file from the RDC, we used a Sleuthkit command icat. The icat command requires three parameters: the drive, the drive's offset number for its file system, and the inode number of the file sought. The drive offset is the number of bytes to go from the beginning of the drive before it can start looking at data (Venema, n.d.) (the usual offset is 63). The inode number is like an address of the file on the drive; we obtained it from the output of the Fiwalk tool included within Sleuthkit that extracts file metadata. The output was given in the following manner: |182|$OrphanFiles/17383091_mp4_h264_aac[1].mp4|0|0|0|AE10-1003|71262|63|, where the drive is AE10-1003, the inode number is 71262, and the offset is 63.

C. DATA TESTING

Before examining the data inside the file, we examined its metadata. The directory path of the file gives more context as to why the picture is on the drive. For instance, if a picture is a blue square, if we saw that the file was named "buttonBackground.gif" and it was under the iTunes directory, then we can infer that it is an uninteresting software GIF. Also, the size of a file could give clues to the nature of the file. For example, if a GIF file is less than 1 inch tall by 1 inch wide but the size of the file were a large 200 KB, this discrepancy would make this file interesting because it could show signs of tampering.

To examine the contents of the files, we used several software programs. For most GIF, JPEG, and PNG files, we used the Windows Photos application, which is a default on Windows 10 machines. For some of the PNG files that were hard to view with the Photos application, we also used Windows Photo Viewer, which is a slightly older Windows application. For the MP4 files we used Windows Movies and TV application, another default on Windows 10. For the MP3 files we used the Windows Groove Music application, which is also a default. We also tried a few MP3 files with the Windows Media player as well; to make sure that was not a discrepancy between what we heard from the Groove Music program and the Media Player.

Some image files we could not open, allegedly because of errors in the file. The file could have been deleted and then partially written over, errors in transmission could have occurred when the file was being copied, or a user could have tampered with it. These files we did not feel confident in labeling either as interesting or uninteresting. We counted them separately in our summary tables. When a file could be viewed but it appeared to be damaged (MP3 files in particular had this problem), we put an "N" for the errors column and attempted to judge it as interesting or uninteresting.

We also attempted to classify the files that we could observe as either software, entertainment, or personal. The software designation is for files that were intended for software GUIs, and nearly all of them are uninteresting to investigators. The entertainment designation is for files that were made for the purpose of grabbing a user's

attention for reasons other than software functionality, as for example image files for advertising purposes. Also, images that are meant to be a source of humor or education would be considered entertainment-based. Finally, files that were created by the user of the drive, or had some kind of personal attachment to a user, were classified as personal files: for instance, photos of friends or family or homemade videos. Personal files are the most interesting files for most investigations. Examples pulled from one of the trails during the experiments(Tables 1 and 2) are meant to show the recording techniques that were used during the trails.

Table 1. Example Experimentation Table Part 1

Number	Size (Bytes)	Size (Kilobytes)	Personal, Software, or Entertainment -based (P,S,E)	Errors (Y/N)	Interesting/ Uninteresting	Comments/File Path
1	21924	21.41015625		y		SONGS/mohamadrafi/026 Taarif Karoon Kya Uski.mp3
2	7708556	7527.886719	E	n	I	my music/Nosy Neighbor .mp3
3	66594	65.03320313		y		MOVIES/Side A(Don't Delete)/KisiNeBhi(DilAsH).mp3
4	152	0.1484375		y		DEKHA HAI MAINE [remix].mp3
5	21632	21.125	S	n	U	Documents and Settings/samrat/Local Settings/Temporary Internet Files/Content.IE5/65TIZU54/sfx_ harvest_crop_03[1].mp3
6	5943254	5803.958984	E	n	I	RECYCLER/S-1-5-21-861567501-963894560-725345543-2692/Dd23/課程講解 MP3/D01.mp3
7	4102144	4006	E	n	I	Documents and Settings/SUPANEEWAN/My Documents/My Music/SONGS/13-U CAN' TOUCH THIS (MC HAMMER).mp3
8	5043274	4925.072266	E	n	I	Documents and Settings/Bradley Daniels/Shared/Kanye West - Jesus Walks.mp3
9	3934798	3842.576172	E	n	I	Documents and Settings/Owner/My Documents/My Music/LC/faith hill - she's in love with the boy.mp3|
10	3819	3.729492188		y		Shishumani Songs/MOVIE SONGS/Chameli/JaneKyonHumko.mp3

Table 2. Experimentation Table Part 2

Number of Files that had errors:
23
Number of files marked as Interesting:
25
Number of Interesting Files over 5 KB:
25
Percent of files W/O errors that were interesting:
92.59259259
Percent of Interesting Files that were over 5 KB:
100

When examining the files, several metadata parameters affected whether a file could be interesting. For example, if a file appears to be software-based (which is usually uninteresting) but its size was very large or the file name was strange, then it could be interesting. Also, if the file appears to be entertainment (which is usually interesting) but shows something relatively generic (like a Target advertisement), then it could be uninteresting. The criterion was "would an analyst for a criminal investigation or for intelligence gathering want to see this?" As shown in the tables above, we also kept track of how many files had critical errors on attempting to open them, the number of files that were interesting, and the number of files that we could judge were interesting and were over 5 KB large.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS

We went through the files in our sample that were deemed interesting by Rowe's program previously, and then those that were deemed uninteresting. For each, we kept track of how many files we found to be truly interesting, how many we found uninteresting, the size of the files, and the type of files (software, entertainment, or personal).

A. CORRECTNESS OF LABELING IMAGE FILES

We tested 100 GIFs tagged as potentially interesting by Rowe's software, of which 63 had critical errors when we tried to open them, which left 37 to investigate for interestingness. Of the 37 files, none was deemed to be truly interesting to either a criminal investigation or intelligence operation. Also, of the 100 uninteresting files that we randomly chose, 36 had critical errors when we tried to open them, which left 64, of which none was deemed to be interesting.

We chose 50 JPEG files tagged as interesting by Rowe's software, of which 30 had errors when we tried to open them, which left 20 to judge for interestingness. For these 20, none was found to be interesting.

Of the 50 JPEGs originally chosen from those marked as uninteresting by Rowe's software, only four had critical errors, which meant that 46 files could be observed. Of these 46 files, three JPEGs were determined to be interesting. All of these photos were personal, which could be interesting to an investigator. However, these files were later relabeled as uninteresting in the Fiwalk17a results once their directories were recognized, as compared to the results that we were using in the Fiwalk14d file.

We then examined the MP3 files marked as interesting by Rowe's previous experiments. Fifty files were chosen to start, of which 28 had critical errors when we tried to open them. Of the 22 remaining files, none was considered to be interesting.

Of the 50 MP3s originally chosen from those tagged as uninteresting by Rowe's software, 23 had errors when we tried to open them, which left 27 to be investigated. Of the 27 files, none was deemed to be interesting during observation.

We started with 50 MP4 files regarded as interesting from Rowe's previous experiments, of which 13 had errors on attempting to open them. Of the 37 remaining MP4s, only one, which was a video of someone on a family vacation, was actually interesting.

Of the 50 MP4 files in the random sample that were tagged as uninteresting by Rowe's software, 22 had errors on trying to open them, which left 28 files to be investigated. Of the 28 investigated files, two were deemed as interesting. One video was of a person on vacation getting their baggage offloaded from a bus, and the other was of some people doing tricks on motorcycles. Both videos appear to be homemade, and personal to the user.

We were only able to open three of the 50 PNG files in our sample. The other 47 files had critical errors, and the three that we were able to observe ended up being uninteresting to an investigation.

Next, we investigated the PNG files that were ruled-out as uninteresting by Rowe's experiments. In contrast to the previous group of PNG files, however, only 14 files had critical errors in them, which left 36 files to be judged for interestingness. Of the 36 files, none was found to be interesting.

B. ERRORS ON TRYING TO OPEN FILES

Many files in the experiment caused errors upon trying to open them, which prevented further investigation of them. About 60% of the corpus is deleted files, which could provide some insight as to why many of our files were nonfunctional, because files that get deleted from a hard drive often get overwritten with new input data at the front, which could impede proper display. The Sleuth Kit, however, can still retrieve these deleted files, which is why we saw them in the first place.

To investigate further, we checked all of the files in our Interesting GIF sample, to see if there was any correlation between the deletion of a file and whether or not it was openable. Of the 100 GIFs in the trial, 68 were deleted. However, nine files could be opened that were deleted, and three files could not be opened but were not deleted. In summary, a deleted file was more likely to be unopenable, and unopenable files were more likely to have been deleted, but neither tendency was guaranteed.

Another reason why some files were unopenable could be software age. The corpus holds files that are up to 20 years old, and most modern software systems may not support them anymore. While we tried using multiple software programs to view the files, all of the programs we used were made for relatively modern file types. Given more time, we could investigate using legacy software to view some of the files that we could not open during this experiment.

C. RECALL AND PRECISION

Tables 3 and 4 show the recall and precision values of the trials conducted for this thesis. Precision shows the amount of true positives found divided by the sum of true positives and false positives. For instance, if the trial were being conducted on MP4s that were found to be interesting by the Rowe's prior experiments, then the precision would be the number of files we found to be interesting of those labeled as interesting by the software, divided by the total number of files labeled as interesting by the software. Recall is the number of files found to be interesting of those labeled as interesting by the software, divided by the total number of files we judged to be interesting in the software. A high precision value for uninteresting files is important since it means software did not erroneously mark many interesting files as uninteresting. A high recall value for interesting files is also desirable. High precision values for both interesting and uninteresting files are not as important since a low value just means some extra work for an analyst.

Not many files in our sample were truly interesting. Only the MP4 files had a one. The uninteresting files included five interesting files, three of which we later confirmed were errors due to the later analysis in the Fiwalk17a results.

Table 3. Precision and Recall of Interesting Experiments

Experiment	Number of Files Examined	Number of Files Found to be Interesting	Precision	Recall
GIF	37	0	0	1.0
JPEG	20	0	0	1.0
PNG	3	0	0	1.0
MP3	22	0	0	1.0
MP4	37	1	0.027	0.973

Table 4. Precision and Recall of Uninteresting Experiments

Experiment	Number of Files Examined	Number of Files Found to be Uninteresting	Precision	Recall
GIF	64	64	1.0	1.0
JPEG	46	43	1.0	0.935
PNG	36	36	1.0	1.0
MP3	27	27	1.0	1.0
MP4	28	26	1.0	0.929

VI. CONCLUSION

A. SUMMARY

In our experiments conducted on GIFs, JPEGs, MP3s, MP4s, and PNG files, we found only a few truly interesting files for standard forensic purposes. Our results show that the rules used by Rowe's experiments work well for excluding files that are interesting but labeled at uninteresting, but there are still many uninteresting files marked as interesting when there is uncertainty about them.

B. FUTURE WORK

When looking at the files, we could draw some conclusions that could prove useful for future work. First, the interesting files that were found were rather large, at least over 1 KB. While Rowe's rules already have a File Size Rule to check for interestingness, its threshold is only about six bytes. If that rule was changed to account for file format, however, then the threshold could be increased to 1 KB for the five file formats covered in this thesis. The MP3 and MP4 thresholds could be increased even more because they are usually much bigger than the other file formats.

Another possible pattern could be observed with the file type. All of the files labeled as truly interesting in this thesis were classified as personal. Later experiments could test to see if a rule could be made to look for a file's type based on its directory, and then classify it as interesting if it is personal.

Other future work could focus on the Time Created, Last Modified, and Last Accessed metadata of a file. For instance, if a file is shown to have been modified before it was created on the machine, then that could be a sign of software, and therefore the file would be uninteresting. Another clue for uninterestingness could be that a file was created but never accessed, which again suggests software.

Table 5. Interesting Files and their Sizes

File Type	File Number	Size in Kilobytes
Uninteresting JPEG	7	58.73
Uninteresting JPEG	10	6.24
Uninteresting JPEG	43	7.16
Interesting MP4	13	1289.28
Uninteresting MP4	21	1100.13
Uninteresting MP4	26	4050.72

C. WEAKNESSES OF METHODOLOGY

Some aspects of these experiments could be improved. First, the way of judging files as interesting or uninteresting should be practiced before carrying out the official experiments. While the reasoning for judging a file as interesting or not stayed the same throughout the experiments, our experience for judging whether a file met those requirements changed. For example, during the opening experiments for the GIF files, we ran across a file that was of a certain country's flag. Initially we marked it as interesting. However, after some thought, we recognized that the flag file was probably just a software GIF that the user has never seen before, and would be uninteresting to an analyst for either criminal investigations or intelligence gathering. With any experimental process that requires the investigator to make a judgement call, there will always be a learning curve for gaining enough experience to make accurate calls.

In addition, we marked files that were interesting to both a criminal investigation and intelligence gathering during these experiments, but most such files would be pertinent only to an intelligence analyst. Aside from a few MP4 and MP3 files, there were few files that we would have considered important to a criminal investigation. We do not believe that we found anything illegal on the drives during the experiments. The bulk of the interesting files could only tell an investigator about what a user has an interest in, or who the user's family and friends are. While these pieces of information could be interesting during an investigation, it may not be interesting when the analyst is looking for evidence for a trial. For this reason, we believe the results of our experiments could be different if we did it for the purpose of helping criminal investigations only. Perhaps another option for future work could be to make custom file-scanning software for finding interesting files in a specific type of investigation, such as a criminal investigation.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Carrier, B. (n.d.). Description. Retrieved August 14, 2017, from http://www.sleuthkit.org/sleuthkit/desc.php

Chawathe, S. (2012) Fast fingerprinting for file-system forensics. *IEEE Conference on Technologies for Homeland Security*, Waltham, MA, November, 585–590.

Fileformat. (n.d.). GIF file format summary. Retrieved June 23, 2017, from http://www.fileformat.info/format/gif/egff.htm

FileInfo. (2016a). MP4 file extension. Retrieved August 28, 2017, from https://fileinfo.com/extension/mp4

FileInfo. (2016b). MP3 file extension. Retrieved August 28, 2017, from https://fileinfo.com/extension/mp3

FileInfo. (2017a). JPEG file extension. Retrieved August 28, 2017, from https://fileinfo.com/extension/jpeg

FileInfo. (2017b). PNG file extension. Retrieved August 28, 2017, from https://fileinfo.com/extension/png

Kornblum, J. (2006). Identifying almost identical files using context triggered piecewise hashing. *Digital Investigation*, 3, 91–97.

Long, C., & Guoyin, W. (2008). An efficient piecewise hashing method for computer forensics. *IEEE First International Workshop on Knowledge Discovery and Data Mining* (WKDD 2008), pp. 635–638.

Mohaisen, A., & Alrawi, O. (2014, July). An evaluation of antivirus scans and labels. *11th Intl. Conf. on Detection of Intrusions and Malware and Vulnerability Assessment*, Egham UK, pp. 112–131.

Rowe, N. (2015). Identifying forensically uninteresting files in a large corpus. *EAI Endorsed Transactions on Security and Safety, 3*(7), 1–15.

Rowe, N., & Garfinkel, S. (2012) Finding suspicious activity on computer systems. *11th European Conf. on Information Warfare and Security*, Laval, France.

Ruback, M., Hoelz, B., & Ralha, C. (2012) A new approach to creating forensic hashsets. *Advances in Digital Forensics VIII, IFIP Advances in Information and Communication Technology* Volume 383, Pretoria SA, 83–97

Tatham, S. (1997a). Chapter 6: Using PSFTP to transfer files securely. *PuTTY user manual*. Retrieved August 14, 2017, from https://www.ssh.com/ssh/putty/putty-manuals/0.68/AppendixC.html#licence.

Tatham, S. (1997b). Chapter 1: Introduction to PuTTY. *PuTTY user manual*. Retrieved August 14, 2017, from https://www.ssh.com/ssh/putty/putty-manuals/0.68/Chapter1.html#intro

Venema, W. (n.d.). ICAT main page. Retrieve August 28, 2017, from http://www.sleuthkit.org/sleuthkit/man/icat.html

William, D. (2016, March 8). What is a GIF? Retrieved July 10, 2017 from https://smallbiztrends.com/2016/03/what-is-a-gif.html

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California